UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/736,324	12/15/2000	Bela Bodo	040010-907	8709

27045 7590 06/07/2004

ERICSSON INC.
6300 LEGACY DRIVE
M/S EVR C11
PLANO, TX 75024

EXAMINER	
VU, TUAN A	

ART UNIT	PAPER NUMBER
2124	

DATE MAILED: 06/07/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/736,324	BODO, BELA	
	Examiner	Art Unit	
	Tuan A Vu	2124	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *31 March 2004*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-4(5-6 canceled)* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-4* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. This action is responsive to the Applicant's response filed 3/31/2004.

As indicated in Applicant's response, claims 5-6 have been canceled. Claims 1-4 are pending in the office action.

Claim Rejections - 35 USC § 101

2. 35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

3. Claim 1 is rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

> The Federal Circuit has recently applied the practical application test in determining whether the claimed subject matter is statutory under 35 U.S.C. § 101. The practical application test requires that a " useful, concrete, and tangible result" be accomplished. An "abstract idea" when practically applied is eligible for a patent. As a consequence, an invention, which is eligible for patenting under 35 U.S.C. § 101, is in the "useful arts" when it is a machine, manufacture, process or composition of matter, which produces a concrete, tangible, and useful result. The test for practical application is thus to determine whether the claimed invention produces a "useful, concrete and tangible result".

As per claim 1, the claim recites a method of facilitating the tracing of errors in executable software, said software achieved in a building process, wherein a part of said software building process results in a record, said process comprising storing said record, retrieving path and version number and bundling said path and version number with said executable software. The very fact of bundling a path and version number does not amount to a concrete action being performed via tangible means in the art of using computer technologies to make software deliverable; because bundling can be performed mentally without such tangible computer means, i.e. mentally jumbling up data from the executable code with symbols from the path and version

number. Hence, there is no tangible and concrete result associated with a final useful result that would be needed to implement the claimed method of facilitating error tracing as mentioned above. The claim thus does not fulfill the requirements of the practical application test because bundling 2 concepts or symbolic representations amount to no more than an abstract idea. As such, the claim is rejected for being a non-statutory subject matter.

Claim Objections

4. Claim 1 is objected because in line 11, 'path and version' phrase is associated with 'is' instead of 'are'.

5. Claim 4 is objected to because of the following informalities: the verb 'creates' (line 8) is to be corrected to become 'create'. Additionally, there is a missing 'module' between 'said relocatable' and 'together with said source code' (line 16). Appropriate correction is required.

Claim Rejections - 35 USC § 112

6. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

7. Claim 1, 4 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Claim 1 recites the step of 'bundling said path and version number of the record'. As mentioned in the USC 101 rejection, this action of bundling does not specifically point out what action is actually taking place, because the term bundle is not precise a term to describe how the path and version number can be put together with an executable software. The best interpretation would be that an action of compiling is associating the software executable with

the path and version number, however, anything action suggestive of bundling and reasonably reading on such feature will be used.

Likewise, claim 1 and 4 recite 'path and version number of record' (cl. 1, line 9), or 'path and version of the record' (cl. 4, line 11). It is unclear as to what path of the record it is to retrieve. The claims do not specify a path in terms as to enforce a file pathname, nor an precise environment in which some path is associated with the record; nor does it specify what type of path, e.g. a communication path, repository storage path, registry path, a processing or flow path. This path element would be interpreted as any path taken in a generic sense that is related to the record.

The claims dependent of these claims are also rejected.

Claim Rejections - 35 USC § 103

8. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

9. Claims 1-2 are rejected under 35 U.S.C. 103(a) as being unpatentable over McLain, Jr, USPN: 5,956,513 (hereinafter McLain), in view of Leblang et al., USPN: 5,574,898 (hereinafter Leblang); and further in view of Albrecht et al., USPN: 5,950,011 (hereinafter Albrecht)

As per claim 1, McLain discloses a method for tracing the errors in executable software of a computer controlled system, said software is compiled and linked in a building process,

wherein a number of source-code files stored in a version control system (e.g. col. 3, line 65 to col. 4, line 16 – Note: read configuration file to determine version correspondence is implicitly disclosing version control system), and part of said building process further results in a record (*configuration data file, internal table* – col. 4, line 4-46) which specifies names and versions of source-code files included in said build process, such process including the steps:

storing said record in a version controlling system (e.g. col. 13, line 52 to col. 14, line 14; Fig. 5,6a-e – Note: configuration file linking to header files with version information discloses version controlling system for administrating a record),

retrieving path and version number of the header or library files specified by the record (e.g. *latest version, path used*– Fig. 6a-e; col. 15, lines 3-43); and

bundling said path and version number with said executable software(e.g. col. 11, lines 6-27 – Note: including path and version in MAKE linking environment is equivalent to bundling path and version with executable code).

But McLain does not explicitly specify retrieving the version number and path is actually retrieving path and version of the record itself. But McLain discloses a record being specific to a build and a version control associated with a build control, hence suggest a identification number associated with a build (e.g. *ABC 125* - Fig. 1). Analogous to the concept of providing a specific identification to a release or a build thus as suggested by McLain, Leblang, in a method to use version control within a software build, discloses a record or configuration data pointing to versioned objects or header files (e.g. Fig. 23) similar to McLain's, and further associates a version number and location to a global tree representing a release for check-in/check-out (e.g. *version of directory include* - Fig. 17-18; col. 28, lines 21-47); and encompasses build

information and execution instruction in a script (e.g. *clearmake* - col. 28, line 48 to col. 29, line 43). It would have been obvious for one of ordinary skill in the art at the time the invention was made to provide path and version information to the build record as suggested by Leblang in using directory version and build script, each of which entails path and version associated with a build or release as suggested above, to the build method of McLain. One of ordinary skill in the art would be motivated to do so in such case because providing the administrating of versioned high level record (e.g. versioned directory) and build script as taught by Leblang would enhance the view of a particular build among other builds within a hierarchy of build legacy and further would enable the use of one file, e.g. script to encompass the instructions and retrieving information on versioned elements or files to be included in the build, thus achieve version checking at a higher level that file level granularity and facilitating speedy auditing of a build (Leblang: col. 30, line 47 to col. 31, line 29).

Nor does McLain specify that bundling record path and version number is in such a way that said path and version number is retrievable at the site where the executable is to be used. In addition to the use of script to encompass versioned objects and instructions purported to effect version auditing and a build of a specific version/release Leblang discloses a transparent file distribution system via a network of multiple developers communicating with private workstations (Fig. 2, 7) and use of make file (col. 14, line 49 to col. 16, line 16). Additionally, including a linking-related files pathname and version number in Unix MAKE file was a known concept at the time the invention was made as evidenced in Fig. 5 by McLain. For auditing and version checking purposes, it would have been obvious for one skill in the art to provide a communication paradigm and implement it to McLain's method so that a remote auditing using a

script as suggested by Leblang can be effected such that components path and script version number are included wherein to facilitate error tracing regarding file location or version compatibility. Extending the concept as to include version or pathname to a configuration script as suggested by Leblang' auditing process, Albrecht, in a system having a database-connected station for creating a configuration file and a user station for using such configuration file analogous to that of Leblang, discloses including repertoire storage path (col. 10, lines 53-54) of the very configuration file (or build process record) in the configuration file loaded at the user station for interpretation. In a common paradigm of distributed system where user platforms are heterogenous, it would have been obvious for one skill in the art to provide storage location and version of the configuration record related to the software to be downloaded and used by the recipient stations as suggested by Albrecht; and provide it with the remote configuration checking process as suggested by Leblang so the recipient user of the downloaded software can effectively detect configuration compatibility with its environment and would be able to request for upgrade as suggested by Albrecht (col. 2-4) by just comparing downloaded configuration file with resident configuration.

As per claim 2, McLain does not specify post-processing of the executable file with integrating of path and version number therein. But the limitation to provide path and version identifying the configuration record or the script aimed at effecting the build and the version compatibility check has been addressed in claim 1 above using Leblang's teachings and Albrecht's compatibility checking of configuration file at user' end of downloaded software; and the corresponding rejection as set forth in claim 1 herein applies for the same motivation.

10. Claim 3 is rejected under 35 U.S.C. 103(a) as being unpatentable over McLain, Jr, USPN: 5,956,513; Leblang et al., USPN: 5,574,898; and Albrecht et al., USPN: 5,950,011; as applied to claim 1 above; and further in view of Thomas et al., USPN: 6,460,052 (hereinafter Thomas).

As per claim 3, McLain does not specify storing the executable in a PROM. However, Leblang suggests network multi-station communicating of auditing script and Albrecht teaches about network receiving and loading of configuration file with version/path for compatibility checking (re claim 1). In small devices known to be used a processing system in the internet, the storage resources therein being a limiting factor was a known concept at the time of the invention. As an evidence to that fact, Thomas discloses storing of versioned files in a multi-developer similar to that of Leblang, and further discloses storing of code in PROM within a wireless network communication system (e.g. col. 15, line 41 to col. 16, line 14). It would have been obvious for one of ordinary skill in the art at the time the invention was made to provide distributing of software as suggested by Albrecht and Leblang for providing executable at the receiving machine/device so that the executable can be stored in a PROM as suggested by Thomas because the programmable memory unit would accommodate for the lack of storage resources so well-known in small devices such as observed above and illustrated via the wireless communication as suggested by Thomas.

11. Claim 4 is are rejected under 35 U.S.C. 103(a) as being unpatentable over McLain, Jr, USPN: 5,956,513; Leblang et al., USPN: 5,574,898; and Albrecht et al., USPN: 5,950,011; and further in view of Hammond, USPN: 5, 974,470 (hereinafter Hammond) and Luu, USPN: 5,860,012 (hereinafter Luu).

As per claim 4, McLain discloses a method for tracing the errors in executable software of a computer controlled system, said software is compiled and linked in a building process, wherein a number of source-code files stored in a version control system (e.g. col. 3, line 65 to col. 4, line 16 – Note: read configuration file to determine version correspondence is implicitly disclosing version control system), and part of said building process further results in a record (*configuration data file, internal table* – col. 4, line 4-46) which specifies names and versions of source-code files included in said build process, such process including the steps: compiling source code into object files and linking said files (Fig. 2, 3A), wherein said compiling and linking steps also creates a record specifying names and versions of used source code files (*configuration data file, internal table* – col. 4, line 4-46);

storing said record in a version controlling system (e.g. col. 13, line 52 to col. 14, line 14; Fig. 5,6a-e – Note: configuration file linking to header files with version information discloses version controlling system for administrating a record),

retrieving path and version number of the header or library files specified by the record (e.g. *latest version, path used*– Fig. 6a-e; col. 15, lines 3-43); and

creating an object file with including said path and version number (e.g. col. 11, lines 6-27).

But McLain does not specify linking the source files into a relocatable module. But in view of McLain's teaching about locating of source files or header file during the linking process using a configuration file (e.g. Fig. 2, 3A, 6A-D), the limitation as to link files into a relocatable module is implicitly disclosed because if a linked file is not relocatable subsequent linking using such file would not be able to achieve the generation of the object code.

Nor does McLain disclose retrieving path and version of the record thus stored. But this limitation has been addressed in claim 1 above using Leblang's and Albrecht's teachings.

Nor does McLain disclose creating and compiling a source code file where path and version number are defined as global variables; nor does McLain disclose, after such compiling, linking the relocatable module with the source file wherein path and version are defined as global variables.

The limitation to incorporate path and version of the configuration record or script in the receiving end station has been addressed in claim 1 above. But neither McLain, Leblang, and Albrecht explicitly disclose compiling a source code file where said path and version number are global variables and link such file with the relocatable module. The reading of configuration file by Albrecht suggests the global declaration of path and the linking using a Makefile by Leblang suggests the global VERSION of a given script. Further, in the context to support the executing environment that receives the software transferred for activation therein as suggested by Albrecht or Leblang, Hammond, in a system to load dynamic linked libraries in a window machine using version compatibility checking analogous to McLain's or Leblang error checking method, discloses global variables stored in a database to enforce rules for building the module and invoking the functions for activating the dynamic libraries, wherein such global variables include path information of functions needed for the build (e.g. col. 14, line 24 to col. 15, line 29). Further, providing a package including code to that can parse and verify correctness of path of configuration files or components version is disclosed via, for example, Luu's method. Luu, in a method having a client-server paradigm as suggested by Leblang or Albrecht, discloses a package integrating excutable code with post-installation functionalities similar to a Makefile

script to check path of different configuration files (see Appendix col. 13-66). In view of the teachings by Leblang/McLain combined with Albrecht to include path and version information of the auditing script or configuration record to the version compatibility and build, it would have been obvious for one of ordinary skill in the art at the time the invention was made to implement the pathname and version data as global variables imparted to the global record or configuration script as taught by McLain(and further enhanced by Leblang and Albrecht) or to the downloadable source code or package to be activated and loaded such as suggested by Hammond or Luu from above. The reason and motivation for this is that these path and version data would be the uppermost piece of information to parse in the source code in order to have an immediate and starting point to retrieve instructions to retrieve linking files, system configuration and dynamic libraries as intended by McLain, Leblang or Albrecht, especially when the build process resulting package is distributed to the user in a heterogenous network as suggested by Albrecht or wherein the need for an administrator can be obviated according to Luu (col. 1, lines 51-56); and that resolving incompatibilies from reading global variables packaged in a installation executable can alleviate further resources in small recipient systems where execution resources are limited.

Response to Arguments

12. Applicant's arguments filed 3/31/2004 have been fully considered but they are not persuasive. Following are the Examiner's observations.

(A) Applicant has submitted that McLain and Leblang disclose an environment similar to a CASE system and there would be no metadata in e.g. an embedded system when outside of such CASE environment, and that with the invention such embedded system can reduce some

administrative load when outside of the CASE context (Appl. Rmrks pg 4 bottom, pg. 5, middle). The claim does not recite CASE environment nor is it recited in ways to preclude a computer aided software engineering environment even when the configuration file is parsed for activation of code installation or auditing of components being built, such CASE believed to be indispensable to all versioning of software build. Nor does it recite instances where an embedded device can obviate any burden of some non-CASE administrative activities. The claim does not provide explicit description as to distinguish what is achieved when the path and version are read from the configuration script as mentioned in the reference cited from what is claimed as 'in such a manner that ... path and version ... retrievable at the site ...' because the rejection has clearly shown that Albrecht suggests checking of path inconsistency when receiving the configuration file at the user's end. Besides, parsing path and version, especially the version, from downloaded configuration file in order to perform installation and activation of software was a known concept in the art of software distribution. And the reason why it would have been obvious to include path and version for such parsing has been set forth in the rejection. The claim does not explicitly recite in what way the site that receives the stored record path and version information performs the error tracing; nor does it mention about optimization or correctness checking that would otherwise require additional administration burden. In conjunction with the unclear element recited as 'bundling', the step of retrieving from a bundled data is even more obscure as to how such retrieving can be done so to help the error tracing as claimed. For one skill in the art, such lack of disclosing of the essential steps as to implement the proclaimed 'tracing of errors' method amounts to or can lead to a lack of enablement, the

merits of which, at worst, can potentially be subject to rejections prescribed by the Office using specific statutes.

(B) Applicant has submitted that Hiller does not remedy to McLain and Leblang when the environment swiches to outside the CASE (Appl. Rmrks, pg. 6-7). The current rejection has rendered these arguments moot.

(C) Applicant has submitted that McLain, Leblang, Hiller, and Hammond fail to establish prima facie (Appl. Rmrks, pg. 7, bottom, pg. 8). The current rejection has used Hammond and Luu to address the global variables declaring database storage information of files being built and to package executable to check path compatibility of configuration files in a post-installation environment. A prima facie has now being re-established making the arguments moot.

Conclusion

13. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Tuan A Vu whose telephone number is (703)305-7207. The examiner can normally be reached on 8AM-4:30PM/Mon-Fri.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kakali Chaki can be reached on (703)305-9662.

Any response to this action should be mailed to:

Commissioner of Patents and Trademarks

Washington, D.C. 20231

or faxed to:

(703) 872-9306 (for formal communications intended for entry)

or: (703) 746-8734 (for informal or draft communications, please consult Examiner before using this number)

Hand-delivered responses should be brought to Crystal Park II, 2121 Crystal Drive, Arlington. VA. , 22202. 4th Floor(Receptionist).

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

VAT
May 26, 2004

KAKALI CHAKI
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100